INTREPID database, file and data structures (R05)

Top

This appendix describes in detail the database, directory and file structure of INTREPID software and data

In this chapter:

The INTREPID directory structure

Parent topic: INTREPID database, file and data structures (R05)

The files belonging to the INTREPID package are normally stored within the following directory structure. You can locate the directory **`intrepid/`** and its subdirectories anywhere in your directory tree.

`intrepid/`	the main INTREPID directory
`bin/`	contains all machine-dependent files and script files for starting INTREPID modules
`classes/` **`images/`**	class and image libraries
`config/`	INTREPID configuration files and the security lock file
`config/calibration_spectra/`	radiometrics calibration data
`help/` **`manuals/`**	Help files and documentation
`algorithm/` **`lynxdata`**	Links with other software
`gdads/`	GDADS files and sample data
`home/`	(UNIX) sample login environments
`sample_data/cookbook/` **`sample_data/examples/`** **`sample_data/guided_tours/`**	Cookbook datasets Example data Guided Tour data
`proj/`	datum and projection parameter files
`temp/`	directory for INTREPID temporary files
`kernel/`	kernels for Spatial Convolution
`filters/`	filter definition files
`sample_data/examples/ddf/`	DDFs for Import
`lut/`	lookup files
`form/`	form files
`font/`	INTREPID font descriptions
`form/`	forms for the Hard Copy Composition tool

Temporary directories

Parent topic: **INTREPID database, file and data structures (R05)**

INTREPID uses a temporary directory on your hard disc for writing temporary files during processing. The path of the temporary files directory is specified in **`install.cfg`** The default path is ***`install_path`*`/temp`** (where ***`install_path`*** is the location of your INTREPID installation. See Configuration (`.cfg`) files, `menu` and `.intrepidlock` for more information about **`install.cfg`**.

When you are printing data under *Windows* the operating system controls the temporary files used in the printing process. These files, therefore, are stored in the system temporary directory (e.g., **`c:\temp`**).

INTREPID file names

Parent topic: **INTREPID database, file and data structures (R05)**

Generally speaking, you can use any name that is acceptable to your operating system for INTREPID datasets but we recommend that you note the following:

- We recommend that you do not use spaces in file, folder or vector dataset field names. Use '_' instead.
- INTREPID uses the '.' character for cetain purposes. We recommend that you don't use it in file, folder or vector dataset field names.
- INTREPID is file-compatible between *UNIX* and *Windows*. This means that you can create and process the same datasets with either version of INTREPID. There are some small issues to remember, however.
- *Windows* and *Mac OS* are case insensitive but case preserving. For example if a dataset is called **`MyGrid.ers`**, it will always show as **`MyGrid.ers`**. However, if you type **`mygrid.ers`** when specifying the dataset, INTREPID locates **`MyGrid.ers`** for you.
- *Linux* and *Solaris* are case sensitive and case preserving. This means that theoretically you could have two datasets existing side by side, one called **`MyGrid.ers`** and the other called **`mygrid.ers`**.
- To minimise risk of confusion, it may be easier to avoid capital letters and spaces in file names.

INTREPID file types

Parent topic: INTREPID database, file and data structures (R05)

The following table contains a complete list of file types associated with INTREPID.

* From INTREPID version 3.7, the INTREPID standard information (**.isi**) file replaces these files.

† Early INTREPID file, now obsolete

File Name/ Extension	File Type	Associated with	Format
INTREPID software files			
***.exe** ***.dll** ***.so** ***.jar**	program (executable) file and library	INTREPID	Binary
***.hlp**	help text file for online help (currently being replaced with more advanced on-line help)	INTREPID	ASCII
menu ***.mnu**	Project Manager menu specifications	INTREPID	ASCII
INTREPID project files			
HISTORY	history (audit trail) file for project	project	ASCII
ntout **nt.*username*.log**	diagnostic report produced during process *(Windows only)*	working directory	ASCII
INTREPID vector dataset files			
***.isi**	INTREPID standard information file From version INTREPID 3.7, it replaces the files marked * in this table.	vector dataset (outside directory)	ASCII
***.PD** (vector header)	Header file for a vector dataset*	vector dataset (outside directory)	Binary
***..DIR**	File Marking the presence of a vector dataset. Dataset directory has the same name.	vector dataset (outside its directory)	Binary
INDEX..INDX	File Marking the presence of the index file	vector dataset	Binary
INDEX.PD	Index file synchronising the data in all of the **.PD** files. DO NOT DELETE!	vector dataset	Binary
SurveyInfo	survey information file*	vector dataset	ASCII
***.PD.vec**	header file for a field containing datum, projection and statistical information*	vector dataset	ASCII
***.PD** (vector field)	data file for the index or for one field of a vector dataset	vector dataset	Binary

File Name/ Extension	File Type	Associated with	Format
***..LINE**	file marking the presence of a field	line dataset	Binary
***..PNT**	file marking the presence of a field	point dataset	Binary
***..POLY**	file marking the presence of a field	polygon dataset	Binary
AREA	file for polygon datasets (old version)†	old polygon dataset	Binary
***.rpt**	ASCII columns import report*	line dataset (outside its directory)	ASCII
format	ASCII columns or Geosoft XYZ export specifications*	vector dataset	ASCII
Vector dataset files compatible with INTREPID			
*.gdb	Vector dataset in Geosoft format (instead of **..DIR** and the dataset directory)	vector dataset	Binary
***.isi**	INTREPID creates a **.isi** file for all datasets that it accesses (except for *Oracle* databases). See the explanation of **.isi** above.	vector dataset	ASCII
*.jdbc	Connection to *Oracle* database (instead of **..DIR** and the dataset directory) Includes information normally stored in a **.isi** file.	vector dataset	ASCII
INTREPID grid dataset files			
***.isi**	INTREPID standard information file From version INTREPID 3.7, it replaces the files marked * in this table.	dataset	ASCII
***.ers** ***.PD.ers**	Header file with *ERMapper* grid information. Use this to load and save an INTREPID grid dataset	grid dataset	ASCII
***.PD** ***.**	Grid data file	grid dataset	Binary
***..GRID**	File marking the presence of a grid dataset*	grid dataset	ASCII
***.FFT**	Fast Fourier Transform and spectral domain grid filter history (audit trail) file	grid dataset	ASCII
Grid dataset files compatible with INTREPID			
***.grd**	Grid dataset in Geosoft format (instead of **.ers**)	grid dataset	Binary
***.cdf**	Grid dataset in *NetCDF* GMT format (instead of **.ers**)	grid dataset	Binary
***.nc**	Grid dataset in *NetCDF* format that is compatible with JetStream (instead of **.ers**)	grid dataset	Binary

File Name/ Extension	File Type	Associated with	Format
***.isi**	INTREPID creates a **.isi** file for all datasets that it accesses (except for *Oracle* databases). See the explanation of **.isi** above.	vector dataset	ASCII
INTREPID auxiliary files			
***.cfg**	configuration specifications	INTREPID	ASCII
***.proj**	projection parameter files	INTREPID	ASCII
***.datum**	datum parameter files	INTREPID	ASCII
projectionDB	projections database file	INTREPID	ASCII
***.lut**	colour lookup table	INTREPID	ASCII
***.ker**	convolution kernel definition	INTREPID or your data	ASCII
***.fdf**	filter definition file	INTREPID or your data	ASCII
***.frm** in directory specified by you	formula file containing **if – then – else** specification for modifying data	INTREPID or your data	ASCII
.frm** in directory ***install_path **/form**	(where ***install_path*** is the location of your INTREPID installation) Window specifications for Hard Copy Composition (Do not modify these files or store other files in this directory.)	INTREPID	ASCII
***.asc**	Radiometrics calibration spectra	INTREPID	ASCII
***.DDF**	Data Description File	your import data	ASCII
***.job**	task specification file	your data	ASCII
***.spectrum.asc**	report of Fourier coefficients with cycles per metre, log of power and average depth estimate after reverse FFT	grid dataset	ASCII
***.map**	hard copy specification in MAPCOMP language	your hard copy data	ASCII
***.leg**	legend specification file for Hard Copy Composition	your hard copy data	ASCII
.leg** in directory ***install_path **/lut**	(where ***install_path*** is the location of your INTREPID installation). Field values lookup for: • substituting values in ASCII columns import or • associating with a field to display lookup results instead of stored value	a DDF file or a vector dataset field	ASCII
BMRStandard Channels	import information for AGSO vector data import	INTREPID	ASCII

Vector dataset file and directory structure

Parent topic: INTREPID database, file and data structures (R05)

A vector dataset consists of a directory, a marker file ending with **..DIR** and an INTREPID standard information (**.isi**) file[1].

Inside the directory there are

- Two index files:
  - A marker file called **INDEX..INDX**
  - A data file for the index called **INDEX.PD**
- A survey information file called **SurveyInfo** (This file is obsolete. See Vector dataset field aliases for details.)
- Three files for each field of the dataset:
  - A marker file for the field with the field name and an ending **..LINE**, **..PNT** or **..POLY** depending on the type of dataset
  - A data file with the field name and the ending **.PD**
  - A header file containing information such as datum, projection and statistics with the field name and the ending **.PD.vec**

  Line datasets have field marker files ending in **..LINE**

  Point datasets have field marker files ending in **..PNT**

  Polygon datasets have field marker files ending in **..POLY**

In this section:

1.Datasets earlier than INTREPID version 3.7 have a header file ending with **.PD**. The **.isi** file replaces this file after version 3.7.

INTREPID standard information (.isi) files for vector datasets

Parent topic: Vector dataset file and directory structure

Data in a vector dataset INTREPID standard information (**.isi**) file includes:

- Field aliases
- Statistics
- Datum and projection
- Field and band header information:
  - Data type
  - Precision
  - 'Group by' indicator
- Report formats

See INTREPID standard information (.isi) files.

Vector dataset field aliases

Parent topic: Vector dataset file and directory structure

You may wish to use a variety of file names for the fields and have several versions of some fields (e.g., uncorrected and corrected X and Y data). For this reason INTREPID uses a system of aliases for file names. INTREPID may require a number of files to take 'roles' in a process. Each 'role' is signified by an alias. The alias system allows you to vary dataset filenames without having to laboriously specify them for each process.

A dataset's INTREPID standard information (**.isi**) file[1] contains a list of aliases representing specific dataset 'roles' (such as location) and the name of the file associated with each alias (the file which 'plays the role').

Many of the INTREPID tools use the **common aliases**— **X**, **Y** (location), **LineNumber**, **Line Type**, **Fiducial** and **FlightNumber**. Some INTREPID tools, for example, Gravity, Tie Line Levelling, and Radiometrics, include other aliases for their own purposes. If a tool uses special aliases, its chapter will contain the relevant details. If you develop your own software to access INTREPID datasets you may use your own aliases.

Most INTREPID modules that access vector datasets require certain alias definitions. If essential aliases are not defined, INTREPID may prompt you for filenames it would normally find in the alias definition.

An alias definition for a field is an **Alias =** statement in the field's **Begin – End** block within the INTREPID standard information (**.isi**) file. (See INTREPID Auxiliary files for a description).

1.Datasets earlier than INTREPID version 3.7 have a survey information file called **surveyinfo** within the dataset directory. INTREPID maintains it for backwards compatibility. Avoid interfering with **surveyinfo**.

Editing aliases

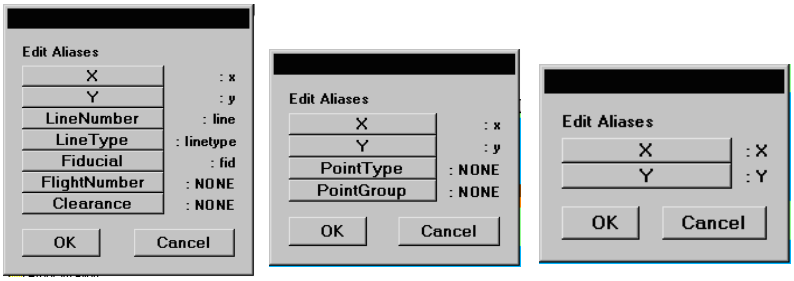
You can edit the aliases of a dataset using a text editor or while using the following tools:

- The Project Manager (See "Managing dataset aliases" in INTREPID Project Manager (T02)),
- The Old Project Manager (See "Editing the dataset aliases" in INTREPID Old Project Manager (T01)),
- The Import tool (See "Editing the aliases" in Importing to INTREPID datasets (T05)),
- The Levelling tool (See "Editing aliases" in Line correction and tie levelling (T30)),
- The Line Filters tool (See "Editing aliases" in Line Filtering (T31)),
- The Multi-channel Spectra Processing and Correction tool (See Multi-channel gamma ray spectrometric processing (C07)).

>> To edit the aliases of a dataset

Note: The Project Manager has a different user interface different to the one described here. For instructions, see "Managing dataset aliases" in INTREPID Project Manager (T02)

1 Select the dataset and display the Dataset Alias Editor according to the instructions for the tool you are using.



2 Choose the alias you wish to change by clicking its button. INTREPID displays a file selection dialog box.

3 Choose the field for the alias. (See "Specifying input and output files" in Introduction to INTREPID (R02) for information about specifying files).

4 Repeat steps 2 and 3 for any other aliases you wish to change.

5 Choose OK when you have finished.

'Group by' fields

Parent topic: Vector dataset file and directory structure

It is useful to be able to organise vector dataset data points into groups.

- Line datasets have one group for each traverse line.
- Gravity Point datasets have one group for each gravity station.

INTREPID creates the groups using a 'group by' field. INTREPID stores data in the order in which it arrives from the Import tool. When you nominate a 'group by' field, INTREPID looks for changes in this field between data points. Whenever the value of the field changes in the import and storage order, INTREPID creates a new group.

During import INTREPID records the group as an entity independently of the value of the 'group by' field. Therefore, as a result of changes to the data, you are able to have adjacent groups with the same 'group by' field value. This is useful for flight path datasets, since you can remove sections of traverse line data and convert the remaining segments to separate lines.

If you have two 'group by' fields during import, INTREPID will create a new group when either of them changes.

If you create a new 'group by' field after INTREPID creates the dataset (you can do this using the Spreadsheet Editor), it can only provide additional data for each group. It cannot have any effect on the group structure of the dataset.

INTREPID records whether a field is a 'group by' field in the an **`GroupBy =`** statement in the field's **`Begin – End`** block within the INTREPID standard information (**`.isi`**) file. The possible values are **`Yes`** and **`No`**.

See "'Group by' fields and field aliases" in Importing to INTREPID datasets (T05) for details about 'group by' field identification during import.

See "`BREAK ON / GROUP BY` line" in The INTREPID DDF format (R08) for the notation identifying a 'group by' field in an import data definition file (DDF).

See Spreadsheet Editor (T15) for information about using 'group by' fields in the Spreadsheet Editor.Merging gravity datasets (T56)

See Merging gravity datasets (T56) for instructions on combining datasets so that data with matching 'group by' field values ends up in the same group in the output dataset.

Traverse line numbers and types

Parent topic: Vector dataset file and directory structure

INTREPID has a convention for assigning codes to different types of traverse lines. The INTREPID tools use these codes to identify the line types. For example, the Gridding tool will use the code to recognise and exclude tie lines from the gridding process (See Old Gridding (T22)). INTREPID stores the line type codes in the field with the alias **LineType** (See Vector dataset field aliases for an explanation of aliases)

You may already have a convention for using line numbers to identify line types. The following table sets out the types of traverse line, a common convention for line number allocation and the INTREPID line type code system.

Line type	Line number convention	INTREPID line type code
Acquisition Line	< 7000 last digit 0	2
Tie Line	>= 7000 last digit 0	4
Repeat Acquisition Line	< 7000 last digit 1 for first repeat traverse, 2 for second repeat traverse, etc.	3
Repeat Tie Line	>= 7000 last digit 1 for first repeat traverse, 2 for second repeat traverse, etc.	5
Ignore this line	n/a	16384
Part of diurnal dataset	n/a	8

If your data does not have a **LineType** field but does have a convention for line numbers you can use the INTREPID Spreadsheet Editor (T15) Spreadsheet tool to derive a line type field (See Chapter 3 Spreadsheet Editor in Volume 2 for instructions).

Multiband vector dataset fields

Parent topic: Vector dataset file and directory structure

If you have a number of Z values for each data point. you can store them in separate Z fields or as a single multiband Z field if required. Using the INTREPID ASCII columns import facility can import your data into either format. See The INTREPID DDF format (R08), especially the section "Multiband fields" in The INTREPID DDF format (R08) for details of the importing specifications and "Importing ASCII Columns data" in Importing to INTREPID datasets (T05) for instructions. Electromagnetic and radiometric data uses multiband fields.

The INTREPID standard information (`.isi`) file includes band information for datasets. See INTREPID standard information (`.isi`) files for information.

Viewing line datasets as point datasets

Parent topic: Vector dataset file and directory structure

>> To modify a line dataset for viewing as a point dataset

Create a file with the extension **..PNT** in the line dataset directory. INTREPID will not look inside this file, so it may have any content.

You can then open the dataset as a point dataset.

If you are using the Flight Path Editor, you can choose Cancel when INTREPID prompts you for the Z fields for point colour and size if required.

Other vector dataset information files

Parent topic: Vector dataset file and directory structure

Vector datasets include a number of auxiliary files besides the INTREPID standard information (**.isi**) file. INTREPID creates and maintains these files for backwards compatibility. From version 3.7, INTREPID transfers all relevant information about a vector dataset to the INTREPID standard information (**.isi**) file. It is unlikely that you would need to examine these files or change them. This section describes the files.

Survey information file SurveyInfo

This auxiliary file contains a complete list of the aliases in the dataset. All of its information is also in the INTREPID standard information (**.isi**) file. If the dataset existed before INTREPID 3.7, INTREPID transfers the **SurveyInfo** data to a new **.isi** file the first time you open the dataset. You can safetly ignore **SurveyInfo**.

Here are some examples of survey information files.

Line dataset: The X location field is called **Easting** and the Y location field **Northing**. The line number field is **LINE**, The line type field is **LINETYPE**, the fiducial field is **FID**, the flight number field is **flight** and the clearance field is **rad_alt**. The survey information file for this dataset would be as follows:

```
Alias Begin
   X            = Easting
   Y            = Northing
   LineNumber   = LINE
   LineType     = LINETYPE
   Fiducial     = FID
   FlightNumber = flight
   Clearance    = rad_alt
Alias End
```

Point dataset: The X location field is called **X**, the Y location field **Y**, the point group field is **station_no**, and the point type field is **point_type**. The survey information file for this dataset would be as follows

```
Alias Begin
   X            = X
   Y            = Y
   PointGroup   = Station_no
   PointType    = point_type
Alias End
```

Polygon dataset: The X location field is called **lat**, the Y location field **long**. The survey information file for this dataset would be as follows:

```
Alias Begin
   X            = lat
   Y            = long
Alias End
```

Field header (.vec) information

Each field in a vector dataset has a header file with the extension **.vec**. The header file has INTREPID auxiliary file structure (See INTREPID Auxiliary files for details). INTREPID generates and modifies the header files as you create and process the dataset fields.

From version 3.7, INTREPID extracts the important information from the **.vec** files and includes it in the INTREPID standard information (**.isi**) file. For example, it extracts statistics, datum and projection. Since the important information is in the **.isi** file, you can generally ignore the **.vec** file.

You can manually edit some header file entries if required.

Here are some examples of header files.

A coordinate field from a projected dataset

```
DatasetHeader Begin
   Version = "4.0"
   DataType = Line
   VectorInfo Begin
      NrOfLines = 36
      NrOfCellsPerLine = 1014
      NrOfBands = 1
   VectorInfo End
   CoordinateSpace Begin
      Projection = "TMAMG54"
      CoordinateType = EN
      Units = "METERS"
      Datum = "AGD66"
   CoordinateSpace End
   FormatInfo Begin
      Order = 6
      Width = 16
      Decimals = 2
   FormatInfo End
DatasetHeader End
```

The **CoordinateSpace** block from a coordinate field of a dataset with geographical coordinates

```
   CoordinateSpace Begin
      Projection = "GEODETIC"
      CoordinateType = LATLONG
      Datum = "AGD66"
   CoordinateSpace End
```

A field associated with a field values lookup file **zones.leg**.

```
DatasetHeader Begin
   Version = "4.0"
   DataType = Line
   VectorInfo Begin
      NrOfLines = 36
      NrOfCellsPerLine = 1014
      NrOfBands = 1
   VectorInfo End
   FormatInfo Begin
      LUTFile = zones.leg
      Order = 3
      Width = 7
      Decimals = 2
   FormatInfo End
DatasetHeader End
```

See Fields associated with lookup tables for a listing of **zones.leg**.

A Z field for which statistics has been reported

```
DatasetHeader Begin
   Version = "4.0"
   DataType = Line
   ...
   RasterInfo Begin
      RegionInfo Begin
      ...
         Stats Begin
            NumberOfBands = 1
            NumberOfNonNullCells = 28356
            NumberOfNullCells = 204
            MinimumValue = 1958.932983
            MaximumValue = 2006.137573
            MeanValue = 1972.159285
            MedianValue = 1972.159285
            CovarianceMatrix = 33.206414
         Stats End
      RegionInfo End
   RasterInfo End
DatasetHeader End
```

DataType = This line describes the type of dataset to which the field belongs (**Line Point Polygon**)

VectorInfo Begin—VectorInfo End This block describes the dimensions of the field. The entries are self explanatory.

FormatInfo Begin—FormatInfo End This block describes the format of the field for display in the Spreadsheet Editor. **Order** specifies the column order position of the field. **Width** specifies the display width for the column. **Decimals** describes the number of decimal places displayed for the field. You can edit the specifications in this block if required.

Field values lookup file specification This consists of a **`LUTFile = filename`** statement. You do not need to specify the full path of the file or its extension, since all such files have a required **`.leg`** extension and must reside in the ***`install_path`*`/lut`** directory (where ***`install_path`*** is the location of your INTREPID installation. Place the specification statement in the **`FormatInfo Begin – FormatInfo End`** block. See Fields associated with lookup tables for further information.

RasterInfo Begin—RasterInfo End If you examine statistics for the field by double clicking it in the Project Manager, INTREPID will save a copy of the statistics report in this block. From this time onwards, whenever the data in the field changes, INTREPID will update this report in the header file.

Vector dataset examples

Parent topic: Vector dataset file and directory structure

Line dataset example

A line dataset called **`ebagoola`** has 7 fields—**`latitude, longitude, linenumber, linetype, flightno, fiducial, magnetic`**. It consists of the **`ebagoola..DIR`** and **`ebagoola.PD`** files and the **`ebagoola`** directory

```
ebagoola..DIR
ebagoola.isi
ebagoola/
  INDEX..INDX
  INDEX.PD
  SurveyInfo
  latitude..LINE
  latitude.PD
  latitude.PD.vec
  longitude..LINE
  longitude.PD
  longitude.PD.vec
  linenumber..LINE
  linenumber.PD
  linenumber.PD.vec
  linetype..LINE
  linetype.PD
  linetype.PD.vec
  fiducial..LINE
  fiducial.PD
  fiducial.PD.vec
  flightno..LINE
  flightno.PD
  flightno.PD.vec
  magnetic..LINE
  magnetic.PD
  magnetic.PD.vec
```

Point and Polygon datasets

Point and polygon datasets have a similar structure to the line dataset illustrated in Line dataset example except that they have **`..PNT`** and **`..POLY`** marker file endings instead of **`..LINE`**

Grid dataset file and directory structure

Parent topic: INTREPID database, file and data structures (R05)

Grid datasets have three files with the same name but different endings:

- INTREPID standard information (**.isi**) file
- A data file with no extension or ending in **.PD**
- An *ERMapper* header file ending in **.ers**

If you have turned on *ARC/INFO* or ERDAS *IMAGINE* access, INTREPID will create grid files with header (**.hdr**), projection (**.prj**) and binary interleaved link (**.PD.bil**) files when you save a grid dataset.

In this section:

INTREPID standard information (.isi) files for grid datasets

Parent topic: Grid dataset file and directory structure

Data in a vector dataset INTREPID standard information (**.isi**) file includes:

- Data and band header information:
  - Data type
  - Precision
- Datum and projection
- Statistics

See INTREPID standard information (.isi) files for a general description of these files.

Grid dataset *ERMapper* header (.ers) files

Parent topic: Grid dataset file and directory structure

Each grid dataset has an *ERMapper* header (**.ers**) file in INTREPID auxiliary file format.

The information in the file enables you to open it in *ERMapper*.

Here is an example of a **.ers** file.

```
DatasetHeader Begin
  Comments="Audit Stamp V3.1c-27/ 5/1996"
   Version = "4.1"
   CoordinateSpace Begin
      Projection = "TMAMG54"
      CoordinateType = EN
      Units = "METERS"
      Datum = "AGD66"
      Rotation = 0:0:0
   CoordinateSpace End
   HeaderOffset = 512
   DataSetType = ERStorage
   DataType = Raster
   ByteOrder = LSBFirst
```

```
      RasterInfo Begin
         CellType = IEEE4ByteReal
         NullCellValue = -5.0E+75
         RegistrationCoord Begin
            Eastings = 739961.150000
            Northings = 8420039.770000
         RegistrationCoord End
         RegistrationCellX = 0
         RegistrationCellY = 0
         CellInfo Begin
            Xdimension = 80.000000
            Ydimension = 80.000000
         CellInfo End
         NrOfLines = 150
         NrOfCellsPerLine = 150
         NrOfBands = 1
         RegionInfo Begin
            Type = Polygon
            RegionName = "All"
            SubRegion = {
                              0                   0
                              0                 150
                            150                 150
                            150                   0
            }
            Stats Begin
               NumberOfBands = 1
               NumberOfNonNullCells = 22186
               NumberOfNullCells = 314
               MinimumValue = -0.2996773720
               MaximumValue = 0.3520531654
               MeanValue = -0.0001314568
               MedianValue = -0.0001314568
               CovarianceMatrix = 0.0005585500
            Stats End
         RegionInfo End
      RasterInfo End
   DatasetHeader End
```

Multiband grid datasets

Parent topic: Grid dataset file and directory structure

If you have several Z values for a grid (e.g., derived from the same vector dataset) you can store them as separate grids or as a single multiband grid. Most INTREPID tools which produce grids as output will prompt you to specify the number of bands in the output grid and the band number for use on this occasion. Band numbers start at 1.

Grid dataset example

Parent topic: Grid dataset file and directory structure

ebagrid.isi	(standard files)
ebagrid	
ebagrid.ers	

INTREPID standard information (.isi) files

Parent topic: INTREPID database, file and data structures (R05)

From INTREPID version 3.7, these files are INTREPID auxiliary files (see INTREPID Auxiliary files for details). Each dataset has one. It contains parameter values for the dataset.

When you first create a dataset, the **.isi** file may contain only a small amount of information. As you use INTREPID tools with the dataset, or examine an aspect of it, such as the aliases or the statistics, INTREPID adds more information.

The **.isi** files totally or partly replace some of the other auxiliary files in a dataset. INTREPID maintains the replaced files for backwards compatibility. If you want to change a property of the dataset, edit the **.isi** file rather than the replaced files.

The following sections describe the structure of **.isi** files and give examples.

In this section:

Outer block of the .isi file

Parent topic: INTREPID standard information (.isi) files

```
isi :=

      "MetaData Begin"
            "Name =" {Resource}
            "Version =" {Integer}
            "UserNotes =" {String.List}
            {ColumnDefinition}
            {ColumnDefinition}
            ...
            {ColumnDefinition}
            {ReportFormatDefinition}
            {Extensions}
      "MetaData End"
```

Blocks within the .isi file

Parent topic: INTREPID standard information (.isi) files

```
The "ColumnDefinition"

ColumnDefinition :=

      {Component|ResourceName} "Begin"
            {SingleValuedItems}
            {BandValuedItem}
            ...
            {BandValuedItem}
      {Component|ResourceName} "End"

================================================================
The "ReportFormatDefinition"

ReportFormatDefinition :=

      "ReportFormat Begin"
            "Name = default"
            "Order =" { FieldName.List }
            "Width =" { Integer.List }
            "Decimals =" { Integer.List }
            "DisplayNullAs =" { String.List }
      "ReportFormat End"

================================================================
The "Extensions"

Extensions :=

      "Extensions Begin"
            ...
      "Extensions End"
================================================================
The "SingleValuedItems"

SingleValuedItems :=

      "GroupBy =" {Boolean}
      "DataType =" {DataType}
      "ByteOrder =" LSBFirst | MSBFirst
      "Comment =" {String.List}
      "ReadOnly = " {Boolean}
      "Bands = " {Integer}

================================================================
```

```
The "BandValuedItem"

BandValuedItem :=

     "Band"{BandNumber} "Begin"
          "Minimum =" {Numeric}
          "Maximum =" {Numeric}
          "Mean =" {Numeric}
          "Variance =" {Numeric}
          "Samples =" {Integer}
          "Nulls =" {Integer}

          "Projection =" {ProjectionName}
          "Datum =" {DatumName}

          "BandId =" {BandId}
          "Unit =" {Unit}

          "Width =" {Integer}
          "Decimals =" {Integer}
          "DisplayNullAs =" {String}
          "Alias =" {Alias}

     "Band"{BandNumber} "End"
```

Example of grid dataset `.isi` file

Parent topic: INTREPID standard information (`.isi`) files

```
MetaData Begin
        Name   = table
        tensor Begin
                Bands   = 6
                band3 Begin
                        Minimum =       1.000000
                        Maximum =       6.000000
                        Mean    =       3.500000
                        Variance=       8.333333
                        Samples = 4
                        Nulls   = 0
                band3 End
        tensor End
MetaData End
```

Example of vector dataset `.isi` file

Parent topic: INTREPID standard information (`.isi`) files

```
MetaData Begin
      Name = BrokenHill
      UserNotes = {
                  "This is a demo dataset"
                  "containing magnetics for the BrokenHill Area"
                  }
      X Begin
            Alias = { X }
            Minimum = 530000.0
            Maximum = 540699.0
            Mean = 535399.4
            Variance = 14599
            Samples = 147031
            Nulls = 0
            Projection =  "TMAMG54"
            Datum = "WGS84"
      X End
      Y Begin
            Alias = { Y }
            Samples = 147031
            Nulls = 0
            Projection =  "TMAMG54"
            Datum = "WGS84"
      Y end
      FID Begin
            Alias = { Fiducial }
      FID End
      LINE Begin
            GroupBy = yes
            ByteOrder = LSBFirst
            Comment = "The Line number"
            Alias = { LineNumber }
      LINE End
      microlevelled Begin
            DataType = IEEE8ByteReal
            ReadOnly = yes
            BandId = mag
            Unit = nT
      microlevelled End

      ReportFormat Begin
            Name = Default
            Order = { X Y FID LINE microlevelled }
            Width = { 12 12 10 6 10 }
            Decimals = { 0 0 0 0 2 }
            DisplayNullAs = { -999 -999 -999 -999 -999 -999 }
      ReportFormat End
MetaData End
```

Projects—file and directory structure

Parent topic: INTREPID database, file and data structures (R05)

For an introduction to the project directory, working directory and history file, see "HISTORY files" in Introduction to INTREPID (R02) and "The HISTORY file location, project directory, working directory" in Introduction to INTREPID (R02)).

Project example

This simple project has some task and hard copy specification files, a grid dataset (**ebagoola)**, a line dataset (**eba_mag**) and a polygon dataset (**region1**).

project1/	project directory
HISTORY	process audit trail file
import_eba.job	a task specification file
grid_eba.job	a task specification file
eba_print.map	a hard copy specification file
ebagoola.isi	) grid dataset
ebagoola	)
ebagoola.ers	)
eba_mag..DIR	) line dataset marker file
eba_mag.isi	) line dataset header file
eba_mag/	) line dataset directory
INDEX..INDX	) line dataset index files
INDEX.PD	)
SurveyInfo	) survey information (obsolete)
latitude..LINE	) a field of the line dataset
latitude..PD	)
latitude..PD.vec	)
...	) more fields
region1..DIR	) polygon dataset marker file
region1.isi	) polygon dataset information file
region1/	) polygon dataset directory
INDEX..INDX	
INDEX.PD	
SurveyInfo	
X..POLY	
X.PD	
X.PD.vec	
Y..POLY	
Y.PD	
Y.PD.vec	

Data Types in INTREPID datasets

Parent topic: INTREPID database, file and data structures (R05)

Grid datasets and vector dataset fields can be of the following types:

- Scalar, including multiband scalar—multiple values of the same type (for example, in 256 channel radiometrics data fields)—see Scalar data types
- Compound, including vector, tensor, structural gology and observed—see Compound data types

Also in this section:

- Background to the compound data types
- Vector and tensor field data coordinate conventions
- Representing compound data types graphically
- Tensor grid dataset structure
- Fields associated with lookup tables
- The INTREPID null value

Scalar data types

Parent topic: Data Types in INTREPID datasets

The following table shows the scalar datatypes listed in this section.

Data type	Description	Length	Auxiliary file notation	DDF Notation
Scalar				
byte	Unsigned 8 bit integer (values 0..255)	1 byte	**Byte**	**byte**
integer (2 byte)	Signed 16 bit integer	2 bytes	**Signed16BitInteger**	**integer*2**
integer (4 byte)	Signed 32 bit integer	4 bytes	**Signed32BitInteger**	**integer*4**
real (4 byte)	IEEE 32 bit floating point	4 bytes	**IEEE4ByteReal**	**real*4**
real (8 byte)	IEEE 64 bit floating point	8 bytes	**IEEE8ByteReal**	**real*8**
logical*	0=false, 1=true	1 byte	**Logical**	**logical**
character*	character string	variable	**Character**	**character**

*If you wish to use logical or character data in a grid dataset, contact our technical support service for advice.

Compound data types

Parent topic: Data Types in INTREPID datasets

Compound data types bundle a set of related scalar values that together make up a meaningful unit of data.

This section describes notation used in the data description files (DDF) (see The INTREPID DDF format (R08)).

The main table in this section describes the compound data types. The immediately following table explains the column headings:

Heading	Purpose
Data type	Name of data type
Components	Names and descriptions of data type components (combined table cells)
Purpose	Purpose of data type or component
Data Type of Component	Scalar data type of component
Auxiliary file & DDF Notation	Name for data type used in DDF and in auxiliary files such as **`.isi`**
DDF format hint	Notation used in square brackets following the field name in DDF to identify the component. See "Field definition lines" in The INTREPID DDF format (R08).

Compound data type definition table

Some of the components listed here are optional or alternative. The context detemines which are present in the import data or the resulting field data

Data type and Components	Purpose	Data Type of Component	Auxiliary file & DDF notation	DDF Format Hints
Date	Storing date information		**`Date_type`** (in DDF) **`Date`** (in **`.isi`**) **`Date`** (in Project Manager)	
Date (in *dd/mm/yyyy* format)		string		**`Date`**
Year		numeric		**`Year`**
Month		numeric		**`Month`**
Day		numeric		**`Day`**
Time (in *hh:mm:ss.s* format		string		**`Time`**
Hour		numeric		**`Hour`**
Minute		numeric		**`Minute`**
Second		numeric		**`Second`**
Julian day (Days since 1st January)		numeric		**`Jday`**
Seconds since midnight		numeric		**`midnight_seconds`**
Complex	Complex numbers		**`Complex`**	

Data type and Components	Purpose	Data Type of Component	Auxiliary file & DDF notation	DDF Format Hints
Real part		real (8 byte)		**R**
Imaginary part		real (8 byte)		**I**
Vector	Components of a 3 dimensional vector		**Vector** (in DDF) **Vector** (in **.isi**) **Vector3d** (in Project Manager)	
Component	3D components of a potential field measurement		**Component** (in DDF) **Components** (in **.isi**) **FieldComponents** (in Project Manager)	
Gradient	3D components of a potential field gradient		**Gradient**	
X component		real (8 byte)		**Vx**
Y component		real (8 byte)		**Vy**
Z component		real (8 byte)		**Vz**
Type		string		**Vtype**
Structural Geology	Observation of structural geology		**Structural** (in DDF) **Geology** (in **.isi**) **Geology3d** (in Project Manager)	
Strike of measurement (angle)		real (8 byte)		**Strike**
Dip of measurement (angle)		real (8 byte)		**Dip**
Type of rock (code or description)		string		**Geo**
X component of measurement		real (8 byte)		**Sx**
Y component of measurement		real (8 byte)		**Sy**
Z component of measurement		real (8 byte)		**Sz**
Tensor	Tensor (examples)		**Tensor**	
Gradients		real (8 byte)		**Txx** **Txy** **Txz** **Tyx** **Tyy** **Tyz** **Tuv**

Data type and Components	Purpose	Data Type of Component	Auxiliary file & DDF notation	DDF Format Hints
Observed	Measurement		**Observed**	
X gradient		real (8 byte)		**Ox**
Y gradient		real (8 byte)		**Oz**
Z gradient		real (8 byte)		**Oz**
Strength		real (8 byte)		**Ostrength** **Omag**
Tensors		real (8 byte)		**Oxx** **Oxy** **Oxz** **Oyx** **Oyy** **Oyz** **Ouv**
Bearing		real (8 byte)		**Obearing**
Type		string		**Vtype**

Background to the compound data types

Parent topic: Data Types in INTREPID datasets

A family of derived classes have been designed to honour all the commonly available airborne geophysical observation packages.

Magnetic gradiometry

Specifically, for magnetic gradiometry systems, the magnetic intensity plus

- Vertical gradient only
- Transverse gradient (wing tip sensors)
- Transverse & longitudinal gradient (wing tip & tail stinger)
- All gradients (full tri-axial system)
- All components of a field
- Tensor gradients.

Moving platform gravity

For moving platform gravity, the vertical component if available plus

- Full tensor gradients (Bell)
- Vertical component plus motion monitors (L&R / ZLS)
- 2 horizontal curvature tensor (Falcon system)
- 3 gravity components (Sander)

With this approach, each variant is delegated the task of enforcing any innate invariant relationships eg tensor and positive definite symmetric, trace invariance, rotational invariance, boost symmetry etc.

This innate behaviour can be relied upon to carry through in any process involving a manipulation of itself with another reading. This greatly assists the development of algorithms which work with all the various systems in a physically consistent way.

Vector and tensor field data coordinate conventions

Parent topic: Data Types in INTREPID datasets

It is conventional to think of the vertical component of gravity as pointing down. On the other hand, Cartesian coordinates are right-handed, with East as X, North as Y and Elevation as Z. This is often called ENU (East North Up). New generation geophysical field data is recorded in all combinations of ENU, NED and END. END is a left-handed system. Bell FTG tensor data has this convention. The IPHT magnetic tensor is ENU. The Earth's magnetic field (IGRF) is often used in NED when the vector form is required.

INTREPID tries to alert users when confusion about this could make a big difference in results. It is, however, important for you to be aware of your tensor data's coordinate system before you try to process it.

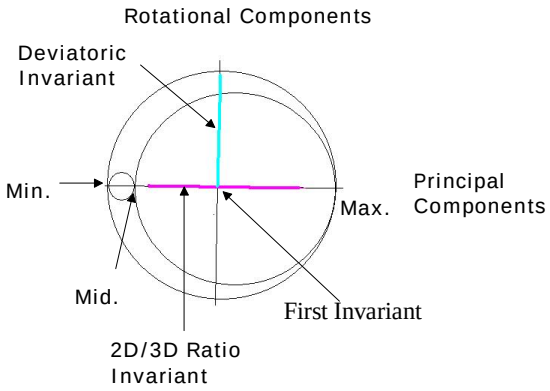
Representing compound data types graphically

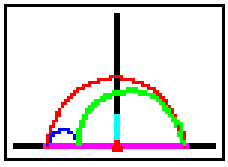
Parent topic: Data Types in INTREPID datasets

INTREPID tools represent some compound data types graphically. This section explains these techniques.

Tensor data

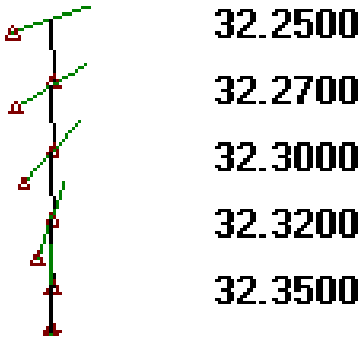
INTREPID tools use the Mohr circle tensor diagram for tensor data. For an explanation of gravity tensors, see http://www.bellgeo.com/tech/technology_theory_of_FTG.html





Moving platform gravity data (Observed data type)

INTREPID tools show this as a green arrow and a black arrow.



The green arrow shows a 'view from above' of the combination of the X and Y components of the gravity field.

The length of the black arrow represents the difference between total gravity magnitude of the data point and the mean total gravity magnitude of the group (traverse line).

The angle of the black arrow corresponds to the total horizontal component of the gravity reading, This shows how far the gravity field is away from vertical.

Since the data type also contains a field strength value, INTREPID shows this value next to the graphic.

Tensor grid dataset structure

Parent topic: Data Types in INTREPID datasets

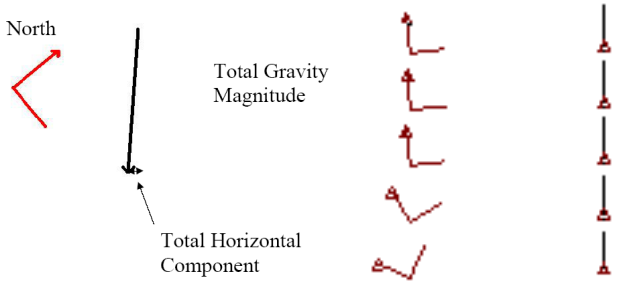
The *ERMapper* (and INTREPID) implementation of a tensor grid uses six bands wih the interleaved by line (BIL) method to store the components, in the order Txx,Txy,Txz,Tyy,Tyz,Tzz. Each band is labelled and has default units of Eotvos.

Moving platform gravity data (Components data type)

We represent moving platform gravity data as a red arrow with two sections at right angles, and a black arrow.

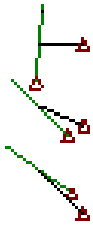
The length of the black arrow represents the difference between total gravity magnitude of the data point and the mean total gravity magnitude of the group (traverse line).

The angle of the black arrow represents the total horizontal component of the gravity reading, This indicates the direction if the gravity field away from vertical.



Structural geology data

Structural geology data types show the angle of a geological feature. INTREPID tools show this as a green arrow and a black arrow. The green arrow shows strike, a 'view from above' of the combined X and Y directions. The black arrow shows a view from the side of the dip.



Complex data

INTREPID tools show complex data as a line segment. The angle and length of the line segment represents real and imaginary parts as a vector.

Fields associated with lookup tables

Parent topic: Data Types in INTREPID datasets

By using integer fields with a lookup table in an auxiliary file, you can create an 'enumerate (ordinal) data type' in an INTREPID vector dataset. A field of this type will have a set of possible values, such as words or phrases, corresponding to integer values. Internally the field is of integer type but INTREPID always displays the corresponding values when you view and edit the data.

For example, data in a dataset could belong to three different zones or to no zone at all. You can have a field which internally can take the values **0..3** but which is associated with a lookup table as follows

```
Lookup Begin
   Name = Zones
   Type = Label
   Label Begin
      Table = {
         0 "Other"
         1 "Zone A"
         2 "Zone B"
         3 "Zone C"
         }
   Label End
Lookup End
```

INTREPID will always display the values **Other**, **Zone A**, etc., depending on the internal value of the field.

Fields associated with lookup files may contain values other than those in the lookup file. INTREPID will use the actual value of the field if there is no corresponding lookup file value.

In the Spreadsheet tool, INTREPID will use the results of the lookup as if they are the actual data stored in the field. See "Fields associated with lookup tables" in Spreadsheet Editor (T15) for instructions.

This feature of INTREPID provides a simple and powerful method of transforming data using Import and Export. You can import data with one set of values, transform it instantly using an associated lookup file, then export it with the new values obtained from the lookup file.

Field values lookup files must reside in the ***install_path*/lut** directory (where ***install_path*** is the location of your INTREPID installation), use the format illustrated above and have the extension **.leg**.

If you import a dataset using a DDF, you can specify a field values lookup file for a field in the DDF. INTREPID will use the values from the lookup file for the data in the field. See "Field definition lines" in The INTREPID DDF format (R08) for details.

INTREPID records the association of a field with a lookup file in the field header (**.vec**) file. See Field header (.vec) information for details. You can specify a lookup file for a field manually by editing the field header file.

The INTREPID *null* value

Parent topic: Data Types in INTREPID datasets

Fields without values contain a standard INTREPID ***null*** value. When INTREPID imports data from other formats, it assigns this value to empty cells. The following table contains the standard values of ***null*** for each data type.

Data type	Value of *null*
Byte	0
Single precision integer	–32768
Double precision integer	–2147483648
Floating point (single and double)	–5.0E75
Logical	0
String	255_{10}

File Masks

Parent topic: INTREPID database, file and data structures (R05)

When INTREPID displays a list of files (including field files) or datasets it finds them by listing certain file types. INTREPID recognises file types using a mask on the file names. INTREPID often uses the **..DIR**, **..LINE**, **..PNT**, **..POLY** marker files to locate vector data for listing. It includes the extensions of compatible dataset formats in the masks. See INTREPID file types for an explanation of the filename extensions.

The following table lists the masks that INTREPID uses. See "Direct access by INTREPID" in INTREPID direct access, import and export formats (R11) for more information about formats that INTREPID can directly access and write to).

File or Dataset Type	Mask
All files	***.***
Vector datasets	***..DIR, *.gdb, *.jdbc**, **.shp**
Point dataset fields	***..PNT**
Line dataset fields	***..LINE**
Polygon dataset fields	***..POLY**
Grid datasets (image formats also available—see "Image formats" in INTREPID direct access, import and export formats (R11))	***.ers, *.grd, *.cdf, *.nc**
History (process audit trail) files	***HISTORY***
Task specification files	***.job**
Hard Copy specification files (MAPCOMP language)	***.map**

INTREPID Auxiliary files

Parent topic: INTREPID database, file and data structures (R05)

INTREPID uses a system of auxiliary files for various purposes such as defining specifications and setting parameters. These are ASCII (text) files with a standard syntax:

- A **Begin – End** block enclosing *keyword=value* declarations.
- Each declaration occupies a new line.
- There may be further **Begin – End** blocks nested within the outer block.
- Lines beginning with # are 'comment' lines for you to include information for people to read about the file. INTREPID ignores the contents of these lines. The **Comment Begin - Comment End** syntax can also be used to define a block of comment lines.
- INTREPID treats text enclosed in **{ }** as a single line even if you have split it into several lines for better readability.

Here is an example

```
Macro Begin
      Name = hpgl
      Device Begin
            Name = hpgl
            Class = Vector
            Coords Begin
                  dpi = 200
            Coords End
#           Colour  Thickness
            Pens = {
                  White   1.0
                  Black   1.0
                  Red     1.0
                  Green   1.0
                  Yellow  1.0
                  Blue    1.0
                  Magenta 1.0
                  Cyan    1.0
                  }
      Device End
Macro End
```

We supply a number of auxiliary files in each category with INTREPID. You can use these files as they are or use them as templates for your own new auxiliary files. You can edit auxiliary files with any text editor.

In this section:

Dataset auxiliary files

Parent topic: INTREPID Auxiliary files

INTREPID datasets have auxiliary files, mainly the INTREPID standard information (`.isi`) file. See Vector dataset file and directory structure and Grid dataset file and directory structure for details.

Process specification files

Parent topic: INTREPID Auxiliary files

These files describe some INTREPID process and can be used to recall a process that has been carried out or specify a process that you require.

Task specification files

Task specification files are text files containing the specifications for tasks to be performed by INTREPID tools.

See INTREPID task specification (`.job`) files (R06) for general information about these files.

History files

History files (called **`HISTORY`** and residing in each project directory) contain an audit trail of processes carried out on the project. Their syntax is the same as task specification files and you can use text from them in task specification files.

See Projects—file and directory structure and INTREPID task specification (`.job`) files (R06) for general information about these files.

Hard copy specification files

If you use the Hard Copy Composition tool or MAPCOMP language you will save your work in **hard copy specification files**. These are text files with standard syntax and names ending in **`.map`**. See MAPCOMP Map Specification Language (R20) for further information.

Datum and projection parameter files

Parent topic: INTREPID Auxiliary files

The datum and projection parameter files specify the projections, ellipsoids and reference points supported by INTREPID. They reside in the directory ***`install_path`*`/proj`** (where ***`install_path`*** is the location of your INTREPID installation).

See INTREPID's supported datums and projections (R09) for further details.

Import/Export format files

Parent topic: INTREPID Auxiliary files

Data Description Files (DDF)

Data Description Files contain specifications for importing ASCII files containing data in columns into INTREPID vector datasets. See The INTREPID DDF format (R08)for a description and "Importing ASCII Columns data" in Importing to INTREPID datasets (T05) for instructions on the use of DDFs.

`BMRStandardChannels` (AGSO vector data)

`BMRStandardChannels` contains field names and other specifications for importing AGSO vector data. See "Importing AGSO vector data" in Importing to INTREPID datasets (T05) for further information.

Export format specification files

These files contain specifications for exporting a vector dataset to an ASCII (text) file or a Geosoft XYZ file. The export format file for a dataset has standard syntax, is called **`format`** and resides in the dataset directory.

See "Export specification files" in Exporting from INTREPID datasets (T07).

Convolution kernel files

Parent topic: INTREPID Auxiliary files

The convolution kernels specify spatial convolution filters for grid datasets. They reside in the directory ***`install_path`*`/kernel`** (where ***`install_path`*** is the location of your INTREPID installation)

See Spatial Convolution Grid Filters (T34) for further details.

Configuration (`.cfg`) files, `menu` and `.intrepidlock`

Parent topic: INTREPID Auxiliary files

The configuration files specify directories, colours, devices, fonts, sizes and configurations for INTREPID. Configuration files reside in the directory ***`install_path`*`/config`** (where ***`install_path`*** is the location of your INTREPID installation). Except for the security lock file they are text files in **`begin – end`** block format. You can edit them according to your requirements.

If you make incorrect changes to some configuration files INTREPID may not function properly. We suggest that you seek advice from our technical support service before editing configuration files if you are uncertain about the changes required.

Colour specifications in these files are in a standard X11 syntax (A **#** followed by three 2-digit hexadecimal numbers (for red, green and blue). For example, black is **`#000000`**, white is **`#ffffff`**, red is **`#ff0000`**, green is **`#00ff00`**, blue is **`#0000ff`**

install.cfg

Specifies :

- Directories used to store INTREPID files, including the directory for temporary storage of work in progress (i.e., the 'temp' directory)
- Aliases for program files so that
  - Old program names still work in your task or hard copy or specification files
  - Programs whose names changed can still work readily after you install INTREPID version 3.1.
- Specifications of the size of your computer's RAM and the size of your swap file (file used for virtual memory)

```
Config Begin
      lutDir    = $(INTREPID)/lut
      kernelDir = $(INTREPID)/kernel
      projDir   = $(INTREPID)/proj
      helpDir   = $(INTREPID)/help
      ddfDir    = $(INTREPID)/ddf
      fontDir   = $(INTREPID)/font
      formDir   = $(INTREPID)/form
      tmpDir    = $(INTREPID)/temp
      ram       = 32
      swap      = 100
#
#     Aliases for Intrepid tools
#
      AdjustSpectra = aspectra
      ConvolveImage = cfilter
      Spreadsheet = dbedit
      decorrugate = decor
      dtrans = dtrans
      Euler = euler
      Export = export
      flightEdit = fedit
      FileManager = fmanager
      Gamadj = gamadj
      GridFilter = gfilter
      GpsNavigationCorrection = gpsnav
      Gridding = gridding
      histogram = histo
      hp_plot = hp_plot
      Import = import
      Levelling = level
      LineFilter = linefil
      MapComposition = mapcomp
      MapPrint = mapprint
      microLevel = mlevel
      ProfileEdit = pedit
      ProjConvert = projconv
      queryindex = qindex
      surveyDistance = sdist
      Stitch = stitch
      subsection = subset
      ulevel = ulevel
      vhbge = vhbge
      vhbgeSpectra = vhbgesp
Config End
```

intrepid.cfg

The default properties file contains window control, size, colour and font specifications for INTREPID generally and for each INTREPID tool.

The following extract from a version of **intrepid.cfg** shows the specifications covering INTREPID as a whole and examples of the specifications for two INTREPID tools.

```
*gui mswin
*PopupWindow*overlay true
*PopupWindow*saveUnder on
*TransientWindow*saveUnder on
*green*background #00ff00
*blue*background #0000ff
*black*background #c2c2c2
*white*background #00ff00
*default #c2c2c2
*wheat*background #c2c2c2
*wheat*foreground #c2c2c2
*foreground #000000
*font *MS Sans Serif*bold*--14*
*background #000000
*title*foreground #333333
*brush_width 0
*double_buffered off
*flat #c2c2c2
*dark #444444
*light #dddddd
*pannerthickness 4
*thickness 4
*framethickness 2
*background #c2c2c2
*editorflat #c2c2c2
*editordark #000000
*editorlight #eeeeee
*editorframethickness 2
...
pedit*font *MS Sans Serif*bold*--14*
pedit*background #c2c2c2
pedit*geometry 650x450
pedit*double_buffered off
pedit*iconName pedit
fedit*font *MS Sans Serif*bold*--14*
fedit*background #c2c2c2
fedit*geometry 650x520
fedit*line_color black
fedit*double_buffered off
fedit*iconName flightEdit

FFT*geometry 760x650
FFT*font *MS Sans Serif*bold*--14*
FFT*background #c2c2c2
FFT*plot0 #0000ff
FFT*plot1 #ff0000
FFT*plot2 #00ff00
```

```
FFT*plot3 #ffaa00
FFT*plot4 #ff00ff
FFT*plot5 #aa3030
FFT*mark_color #ffffff
FFT*axis_color #ffffff
FFT*tick_color #ffffff
FFT*graph_color #444444
FFT*double_buffered off
FFT*iconName gridfilter

level*font *MS Sans Serif*bold*--14*
level*background #c2c2c2
level*foreground #000000
level*double_buffered off
level*active_point_color #0000ff
level*inactive_point_color black
level*point_color #ff0000
level*line_color white
level*mark_color #0000ff
level*axis_color #ffffff
level*tick_color #ffffff
level*graph_color #696969
level*rubber_band #00ff00
level*command #c2c2c2
level*iconName level
level*geometry 800x700

...
```

devices.cfg

The **devices.cfg** file specifies the hard copy devices which INTREPID can use. Each print device that you use must have an entry in this file so that INTREPID can access it. Contact our technical support service if necessary for assistance with specifications for your devices. Here is an example of this file:

```
Macro Begin
      Name = hpgl
      Device Begin
            Name = hpgl
            Class = Vector
            Coords Begin
                  dpi = 200
            Coords End
            Pens = {
                  White   1.0
                  Black   1.0
                  Red     1.0
                  Green   1.0
                  Yellow  1.0
                  Blue    1.0
                  Magenta 1.0
                  Cyan    1.0
                  }
      Device End
Macro End

Macro Begin
      Name = PostScript
      Device Begin
            Name = PostScript
            Class = Raster
            Coords Begin
                  dpi = 72
            Coords End
      Device End
Macro End

Macro Begin
      Name = InterViews
      Device Begin
            Name = InterViews
            Class = Raster
            Coords Begin
                  dpi = 72
            Coords End
      Device End
Macro End
```

```
Macro Begin
      Name = WIN32
      Device Begin
            Name = WIN32
            Class = Raster
            Coords Begin
                  dpi = 300
            Coords End
      Device End
Macro End
```

mapcomp.cfg

This file contains definitions for all elements available in hard copy composition using the INTREPID Hard Copy Composition tool. See Map composition configuration files (R21) for further information and a listing.

menu

The **menu** file contains specifications for the Project Manager menu bar. It resides in the ***install_path*/config** directory (where ***install_path*** is the location of your INTREPID installation). See "Customising the Project Manager menus" in Configuring and using INTREPID (R04) for details.

.intrepidlock

For INTREPID to process your data (i.e., work in other than demonstration mode) you must have an authorisation key file called **.intrepidlock** present in the ***install_path*/config** directory (where ***install_path*** is the location of your INTREPID installation). We provide this file when we issue an INTREPID licence.

Lookup and legend files

Parent topic: INTREPID Auxiliary files

Colour Lookup Tables

The colour lookup tables assign colours to values for displaying INTREPID data. The files reside in the ***install_path*/lut** directory and have extension **.lut** (where ***install_path*** is the location of your INTREPID installation). They are ASCII (text) files with the standard INTREPID **Begin – End** block syntax (see definition of the syntax at the start of this section).

Colour lookup tables should not be confused with field values lookup files which have extension **.leg** and reside in the ***install_path*/lut** directory (see below for details).

The lookup table definition (**.lut**) file contains definitions of four parameters as well as the table. Earth Resource Mapping (publishers of *ERMapper*) developed this format as an open standard.

Version Version number of the lookup table. It must be within "" if it contains spaces.

Name Name of the lookup table. It must be within "" if it contains spaces.

Description Description of the lookup table. It must be within "" if it contains spaces.

NrEntries Number of possible values for the data being represented. This corresponds to the number of lines in the **LUT** matrix.

LUT contains the matrix of colour component levels, one row for each possible value of the data being represented. Column 1 is the level number and columns 2, 3, 4 represent the levels of Red, Green and Blue respectively. The levels range from 0 to 65280.

You must enclose the matrix in braces **{ }**. Each row of the matrix occupies a separate line of the ASCII file. The place for the opening { is on the same line as the words **LUT =** and the } alone on a line under the matrix. The value and levels within the rows are separated by spaces.

Example of colour lookup table definition file

This sample is an extract from a 256 colour lookup table showing the beginning and end of the file.

```
LookUpTable Begin
Version          = "1.0"
Name             = "Rainbow"
Description      = "Cray's Rainbow table"
NrEntries            = 256
LUT                  = {
0          58624      0          65280
1          57600      0          65280
2          56576      0          65280
3          55552      0          65280
.
.
.
252        65280      2816       19968
253        65280      1792       20992
254        65280      768        21760
255        65280      768        21760
}
LookUpTable End
```

Field values lookup files

Field values lookup files associate integer field values with other data such as words. They enable you to have 'enumerate (ordinal)' type data in INTREPID datasets. See Fields associated with lookup tables for details.

Field values lookup files have extension **.leg**, consist of a simple **Lookup Begin – Lookup End** block and reside in the ***install_path*/lut** directory. They should not be confused with colour lookup tables (**.lut**) which reside in the same directory (see above), or hard copy composition legend files, which also have the extension **.leg** (see below).

Hard copy composition legend files

Hard copy composition legends associate values with attributes such as colour and marker size in hard copy compositions. These are text files with standard syntax and names ending in **.leg**. See Map legend files (R23) for details.

INTREPID licence data

Parent topic: **INTREPID database, file and data structures (R05)**

Use the INTREPID Licence Manager tool to manage your INTREPID licence. For full instructions about installing INTREPID and applying your INTREPID licence, see INTREPID Installation guide (S05).

This section explains the location of the licence data in the INTREPID installation. Under normal circumstances you do not require this information.

Windows version

To use INTREPID in normal mode you require an authorisation key file called **`.intrepidlock`** to be present in your ***`install_path`*`/config`** directory (where ***`install_path`*** is the location of your INTREPID installation).

Depending on the time we issued your first licence, you may also require an authorisation device ('dongle') attached to your parallel port or your USB port.

UNIX version

To use INTREPID in normal mode you require an authorisation key file called **`.intrepidlock`** to be present in your ***`install_path`*`/config`** directory (where ***`install_path`*** is the location of your INTREPID installation).